Enumerating Some Fractional Repetition Codes

Srijan Anil
Sapient-Nitro
Gurgaon, Delhi, India
Email:sanil@sapient.com

Manish K. Gupta
Laboratory of Natural Information Processing
Dhirubhai Ambani Institute of Information
and Communication Technology
Gandhinagar, Gujarat, 382007 India
Email: mankg@computer.org

T. Aaron Gulliver
Department of Electrical and Computer Engineering
University of Victoria Victoria, BC, V8W 3P6 Canada
Email: agullive@ece.uvic.ca

***Abstract*—In a distributed storage systems (DSS), regenerating codes are used to optimize bandwidth in the repair process of a failed node. To optimize other DSS parameters such as computation and disk I/O, Distributed Replication-based Simple Storage (Dress) Codes consisting of an inner Fractional Repetition (FR) code and an outer MDS code are commonly used. Thus constructing FR codes is an important research problem, and several constructions using graphs and designs have been proposed. In this paper, we present an algorithm for constructing the node-packet distribution matrix of FR codes and thus enumerate some FR codes up to a given number of nodes n. We also present algorithms for constructing regular graphs which give rise to FR codes.**

I. Introduction

The emerging era of cloud computing poses new challenges for researchers to provide reliable and secure data storage. Practical systems for distributed storage include the Hadoop based system [1] used in Facebook and Windows Azure storage [2]. In these distributed storage systems (DSSs), data is stored on n unreliable nodes. Reliability is provided either by replicating the data or using erasure MDS (Maximum Distance Separable) codes. Both of these schemes have drawbacks either in terms of bandwidth, complexity or disk I/O. To overcome these limitations, regenerating codes were introduced by Dimakis et al. [3], and subsequently studied by many researchers [3], [4], [5], [6], [7], [1], [8]. A node failure in such systems can be handled by regenerating the data stored on that node using its peers. This regeneration can be functional or exact. Functional repair allows restoration of the data such that a stored file can be retrieved by contacting any k out of n nodes, where $k < n$. Exact repair allows for the creation of a replica of the data previously stored on the node [4], [9]. Regenerating codes are specified by the parameters $\{[n, k, d], [\alpha, \beta, B]\}$, where n is the number of nodes, k is the number of nodes that need to be contacted to recover a file B, and d is the repair degree (the number of nodes that must be contacted to regenerate data in case of a node failure). The capacity of a node is given by α, and the repair bandwidth for each of the d nodes is β, so the total repair bandwidth is $d\beta$ [9].

The tradeoff in Regenerating codes between the storage capacity and repair bandwidth have given rise to two new classes of codes, namely Minimum Storage Regenerating (MSR) codes and Minimum Bandwidth Regenerating (MBR) codes. MBR codes employ *exact* and *uncoded* data repair. Uncoded repair means that a particular set of d nodes, as listed in the Repair Table of the node, are contacted and one data packet is downloaded from each, thus reducing the repair complexity. MBR codes are formed by the concatenation of an outer MDS code and an inner Fractional Repetition (FR) code. The MDS code maintains the MDS property of the DSS, while the Fractional Repetition codes allow for an uncoded repair process. These concatenated codes are known as DRESS codes (Distributed Replication-based Exact Simple Storage) codes [9], [5]. Many constructions of Fractional Repetition Codes (and hence DRESS codes), are known based on bipartite graph [10], resolvable designs [11], regular graphs [9], [12] and other structures [8].

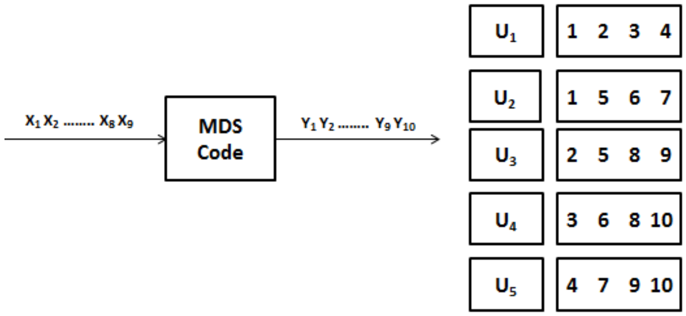


Fig. 1. A DRESS code consisting of an inner fractional repetition code $\mathbb{C}$ having $n = 5$ nodes, $\theta = 10$ packets, replication factor $\rho = 2$, and repair degree $d = 4$, and an outer MDS code.

In this paper, an algorithm for the construction of Fractional Repetition (FR) codes is presented which is based on the incidence matrix of the node-packet distribution. Algorithms are also given for the construction of regular graphs. The rest of the paper is organized as follows. In Section 2, the basics of FR codes and the incidence matrix of the node-packet distribution are given. Section 3 presents an algorithm for the construction of the $n \times \theta$ incidence matrix of the node-packet distribution of an FR code. Algorithms for constructing regular graphs and hence FR codes for $n = \theta$ are presented in Section 4. Finally, Section 5 concludes the paper with some general remarks.

II. Background

Distributed Replication-based Simple Storage (DRESS) codes consist of an inner Fractional Repetition (FR) code and an outer MDS code, as shown in Figure 1. FR codes are formally defined in Definition 1.

Defenition 1. *(Fractional Repetition Code): A Fractional Repetition (FR) code $\mathbb{C}$, with repetition degree ρ, for an (n, k, d) DSS, is a collection $\mathbb{C}$ of n subsets $U_1, U_2, \ldots, U_n$ of a set*

TABLE I. Node-Packet Distribution Incidence Matrix M of Size 5×10 for the FR Code $\mathbb{C} : (5, 10, 4, 2)$ Shown in Figure 1

Node/Packets	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
U_1	1	1	1	1	0	0	0	0	0	0
U_2	1	0	0	0	1	1	1	0	0	0
U_3	0	1	0	0	1	0	0	1	1	0
U_4	0	0	1	0	0	1	0	1	0	1
U_5	0	0	0	1	0	0	1	0	1	1

$\Omega = \{1, \ldots, \theta\}$, *each having size* d, *i.e,* $|U_i| = d$, *satisfying the condition that each element of* Ω *belongs to exactly* ρ *sets in the collection. The code is denoted by* $\mathbb{C} : (n, \theta, d, \rho)$, *and the parameters of* $\mathbb{C}$ *are related by* $nd = \rho\theta$.

Example 2. *Figure 1 gives an example of FR code. Suppose there are 9 packets from* $\mathbb{F}_q$ *(a finite field with* q *elements), and 5 storage nodes. Using an MDS code, the 9 packets are first encoded into 10 packets such that the last packet is the parity packet. Next all 10 packets are replicated twice* ($\rho = 2$), *on the 5 nodes according to the arrangement of the FR code* $\mathbb{C} : (5, 10, 4, 2)$ *in the figure. This code can tolerate 1 failure and the data can be recovered by contacting 4 nodes, hence the repair degree is 4.*

Remark 3. *An FR code* $\mathbb{C} : (n, \theta, d, \rho)$ *can also be characterized by a node-packet distribution incidence matrix* M *of size* $n \times \theta$ *with row weight* d *and column weight* ρ. *For example, the incidence matrix for the FR code* $\mathbb{C} : (5, 10, 4, 2)$ *shown in Figure 1 is given by Table I. The row weight is 4 and the column weight is 2.*

A. Equivalence of Fractional Repetition codes

Two Fractional Repetition codes $\mathbb{C}_1 : (n_1, \theta_1, d_1, \rho_1)$ and $\mathbb{C}_2 : (n_2, \theta_2, d_2, \rho_2)$ are said to be equivalent if

1) The number of nodes and the number of packets in the system are same, i.e., $n_1 = n_2$ and $\theta_1 = \theta_2$. Hence the dimension of the corresponding incidence matrices is the same, i.e., $n_1 \times \theta_1 = n_2 \times \theta_2$.
2) The repair degree and the replication factor are the same, i.e., $d_1 = d_2$ and $\rho_1 = \rho_2$. Hence the corresponding incidence matrices have the same row weight d and column weigh ρ.
3) The same packet distribution can be achieved by simply renaming the packets of one of the codes i.e., if the incidnece matrix of $\mathbb{C}_1$ can be obtained by applying permutations on the rows and columns of the incidence matrix of $\mathbb{C}_2$.

Remark 4. *An incidence matrix of dimension* $n \times \theta$ *defines an FR code with* n *nodes and* θ *packets. The repair degree is* d, *and the replication factor is* ρ. *Now, taking the transpose of this matrix gives a matrix of dimension* $\theta \times n$. *The weight of each row is now* ρ, *and the weight of each column is* d. *This new matrix also satisfies the conditions for an FR code, and corresponds to a code with* θ *nodes,* n *packets, repair degree* ρ, *and replication factor* d.

III. Enumeration of Fractional Repetition Codes using Incidence Matrices

To enumerate the FR codes for a given n, the replication factor ρ can be varied in the range $2 \leq \rho \leq n - 1$, and the repair degree d in the range $2 \leq d \leq n - 1$. In each case, θ can be determined using $nd = \rho\theta$ and the corresponding incidence matrix M of size $n \times \theta$ can be filled such that the weight of each row is d and the weight of each column is ρ to obtain an FR code. Algorithm 1 is given below to fill the incidence matrix with $1's$ and $0's$. Table II summarizes the number of possible FR codes up to length $n = 10$. For larger n, the data can be obtained from http://www.ece.uvic.ca/~agullive/manish/List.html.

TABLE II. The Number of Possible FR Codes for $n = 3$ to 10 Nodes

Number of Nodes (n)	Number of FR Codes
3	1
4	3
5	4
6	10
7	8
8	16
9	19
10	28

Algorithm 1 Generate a node-packet distribution incidence matrix M of size $n \times \theta$

```
Require: $n, d, \theta, \rho$ and an all zero matrix $M$ of size $n \times \theta$
Ensure: $M_{n\times\theta}$ such that $weight(row[M]) = d$
  and $weight(column[M]) = \rho$
  1 : Place a 1 in $d$ positions of the $1^{st}$ row from left to right starting from $m_{11}$ and move to the $2^{nd}$ row.
  2 : In the row, place a 1 in the first column $j, 2 \leq j \leq \theta$ for which the column weight is $< \rho$.
  3 : Compute the weight of all consecutive columns from $j + 1$ to $\theta$. If the minimum weight of these columns is the same, go to Step 4, otherwise place 1's in increasing order of weight until $weight(row) = d$ or the last column is reached. Go to Step 6
  4 : Traversing rows from the top, identify the first row having an entry 1 which corresponds to a 1 in the $j^{th}$ column (determined in Step 2), in the current row.
  5 : Traversing consecutive columns from $j + 1$ to $\theta$ in the current row, place a 1 in the column for which a 0 first occurs in the row identified in Step 4.
  6 : If $weight(row) < d$, go to Step 2 otherwise move to Step 7
  7 : If a next row exists, move to that row and go to Step 2, otherwise Stop.
```

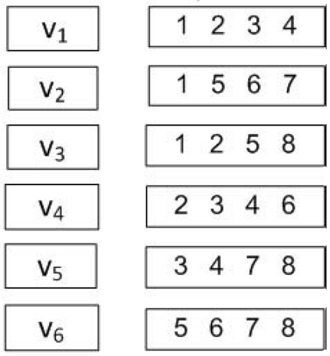
Example 5. *For* $n = 6, d = 4, \theta = 8$ *and* $\rho = 3$, *Algorithm 1 gives the following incidence matrix*

$$M_{6\times 8} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

This matrix gives the FR code $\mathbb{C} : (6, 8, 4, 3)$ *as shown in Figure 2.*

IV. Construction of Regular Graphs

FR codes can be generated using a regular graphs of degree d [9], [12]. Therefore, Algorithm 2 is presented for



v_1	1 2 3 4
v_2	1 5 6 7
v_3	1 2 5 8
v_4	2 3 4 6
v_5	3 4 7 8
v_6	5 6 7 8

Fig. 2. The FR code $\mathbb{C} : (6, 8, 4, 3)$ generated using the incidence matrix in Example 5.

generating regular graphs. We also present Algorithms 3 and 4 for constructing regular graphs based on the approach of filling the incidence matrix to obtain an FR code. To the best of our knowledge, this solution has not been reported in the vast literature on regular graphs. An example is given for each algorithm. The proposed algorithms are constrained to $nd \in 2\mathbb{Z}^+$ and $\rho = 2$. Note that a regular graph of degree d is a graph where every vertex has the same degree d, which is possible only for $nd \in 2\mathbb{Z}^+$.

Algorithm 2 Regular Graph for $nd \in 2\mathbb{Z}^+$, $\rho = 2$ and $d < n - 1$

```
1 : Divide the $n$ vertices into two set of vertices, $U\{u_1, u_2, \ldots, u_{\lfloor \frac{n}{2} \rfloor}\}$ and $V\{v_1, v_2, \ldots, v_{\lceil \frac{n}{2} \rceil}\}$
2 : Construct two cyclic graphs $G_1 : (U, E_1)$ and $G_2 : (V, E_2)$, with $G_1$ enclosing $G_2$
if $n$ is odd then
    Select two vertices $v_i$ and $v_j$ such that edge $\{v_i, v_j\} \notin E_2$
    Add edge $\{v_i, v_j\}$
end if
Select vertices $u_i \in U$ and $v_j \in V$ such that $\deg(v_j) \neq d$
 Add edge $\{u_i, v_j\}$ Repeat for vertex u$_i$ until $\deg(u_i) = \lfloor \frac{d}{2} \rfloor$
Select vertices $u_i, u_j \in U$, such that edge $\{u_i, u_j\} \notin E_1$
 Add edge$\{u_i, u_j\}$
Repeat for vertex u$_i$ until $\deg(u_i) = \lceil \frac{d}{2} \rceil$
Pick vertex $v_i, v_j \in V$, such that edge $\{v_i, v_j\} \notin E_2$
 Add edge $\{v_i, v_j\}$
Repeat for vertex $v_i$ until $\deg(v_i) = \lceil \frac{d}{2} \rceil$
```

Example 6. *Algorithm 2 can be used to generate a d-regular graph for $\rho = 2$ and (a) $n = 4$, $d = 2$ (b) $n = 8$, $d = 4$ (c) $n = 16$, $d = 8$, as shown in Figure 3, and a FR code as shown in Figure 4.*

An adjacency matrix is a matrix depicting the relationship between vertices, showing whether they are connected or not. FR codes can be represented by graphs, where the vertices represent the nodes and the edges represent the packets. These can be interchanged, thus making edges the nodes and vertices the packets. Now for nd even, and $\rho = 2$, graphs can be represented by an adjacency matrix of dimensions $n \times n$. This matrix acts as a basis for generating the incidence matrix of

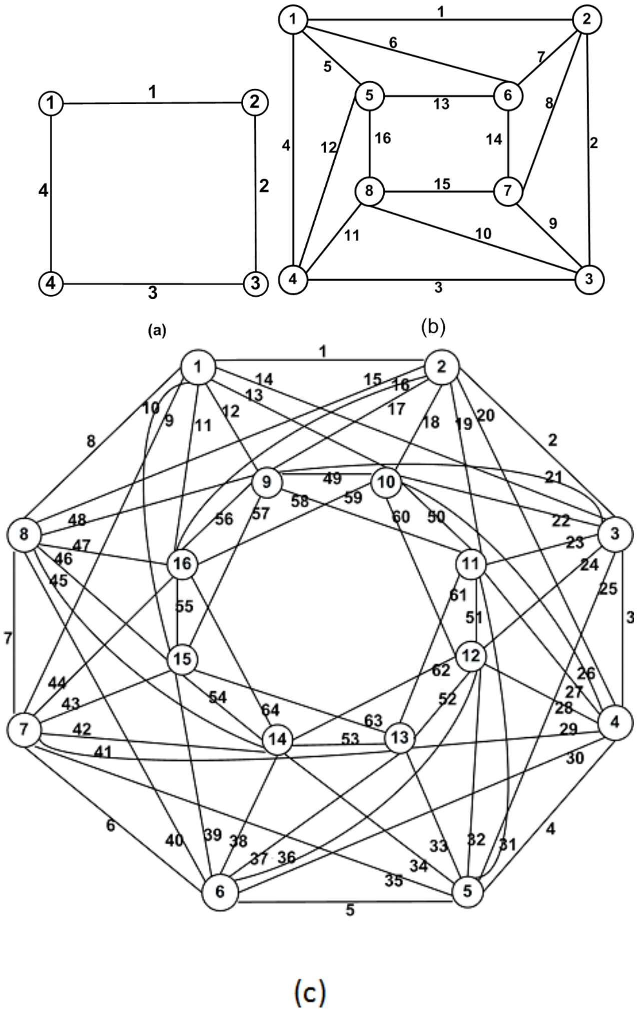
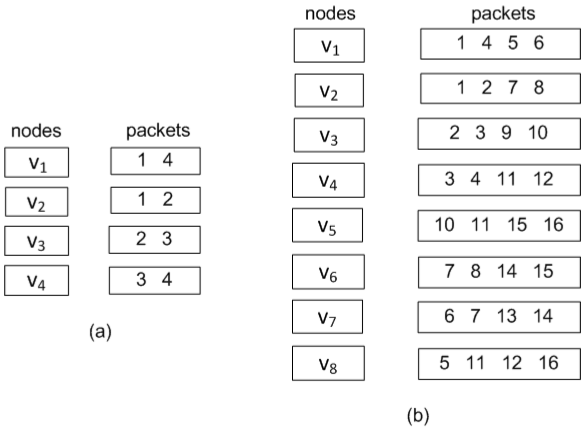


Fig. 3. A regular graph for $\rho = 2$ and (a) $n = 4, d = 2$ (b) $n = 8, d = 4$ (c) $n = 16, d = 8$. The vertices of the graphs are nodes, and the edges originating from them are the packets stored in those nodes. Thus Algorithm 2 generates a d-regular graph which depicts the packet distribution among the nodes as shown in Figure 4



nodes	packets
v_1	1 4
v_2	1 2
v_3	2 3
v_4	3 4

(a)

nodes	packets
v_1	1 4 5 6
v_2	1 2 7 8
v_3	2 3 9 10
v_4	3 4 11 12
v_5	10 11 15 16
v_6	7 8 14 15
v_7	6 7 13 14
v_8	5 11 12 16

(b)

Fig. 4. Node configuration for (a)$n = 4$, $\theta = 4$, $d = 2$, $\rho = 2$ (b) $n = 8$, $\theta = 16$, $d = 4$, $\rho = 2$. This distribution shows that any two nodes have either no packet or 1 packet in common.

the graph. The incidence matrix shows the packet distribution over the n nodes. We present two algorithms to generate the adjacency matrix for parameters ρ, n, d, θ, with constraints ρ=2 and $nd \in 2\mathbb{Z}^+$, where $n = \theta$, row weight d and column weight ρ. This provides an FR code with the same number of nodes and packets.

Algorithm 3 Adjacency Matrix A of Size $n \times n$

```
Require: $n, d, \theta, \rho$ and a null matrix $A$ of size $n \times n$
Ensure: $A_{n \times n}$ such that $weight(row[A]) = d$
    and $weight(column[A]) = \rho$
    1 : Set $a_{12} = 1$ and fill the consecutive entries of first row
    with $(d-1)$ 1's from left to right
    2 : Set the first column as the transpose of the first row
    3 : Move right to left by filling $1's$ such that weight of $i^{th}$
    row is d
    4 : Take transpose of the $i^{th}$ row and fill the $i^{th}$ column
    5 : Increase $i$ by one
    6 : Go to Step 4, if $i < n$.
```

Example 7. *The adjacency matrix for $n = 6$, $d = 4$, $\theta = 8$, $\rho = 3$ generated by Algorithm 3 is*

$$A_{6\times 6} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Algorithm 4 Adjacency Matrix A of Size $n \times n$

```
Require: $n, d, \theta, \rho$ and a null matrix $A$ of size $n \times n$
Ensure: $A_{n \times n}$ such that $weight(row[A]) = d$
    and $weight(column[A]) = \rho$
    1 : For $1 \leq i \leq n$ and $j = n$ to 1
    2 : Update $A[i][j]$ and $A[j][i]$ to 1 $(i \neq j)$ such that weight
    of $i^{th}$ row $= d = \rho$.
```

Example 8. *The adjacency matrix for $n = 6$, $d = 4$, $\theta = 6$, $\rho = 4$ generated by Algorithm 4 is*

$$A_{6\times 6} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

V. Conclusion

In this paper, several algorithms have been presented for constructing FR codes. Algorithm 1 is a general construction technique which for any value of n calculates the possible values of d, ρ and θ, and then generates the corresponding node-packet matrices. The complexity of the algorithm is $\Theta(n^3)$. The algorithm has been tested for values up to $n = 100$, and the results have been recorded. This data is available from http://www.ece.uvic.ca/~agullive/manish/List.html. Our aim was to generate a common data storage pattern for any given set of parameters. The algorithm generates a node-packet matrix for each possible value of d, ρ and θ for a range of n. New algorithms were also presented for constructing regular graphs.

Acknowledgment

The authors would like to thank Krishna Gopal Benerjee for useful discussions and Nikhil Agrawal for writing parts of the program for FR code enumeration and drawing some of the figures.

References

[1] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "Xoring elephants: Novel erasure codes for big data," *Proceedings of the VLDB Endowment (to appear)*, 2013.

[2] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in windows azure storage," in *Proceedings of the 2012 USENIX conference on Annual Technical Conference*, ser. USENIX ATC'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 2–2. [Online]. Available: http://dl.acm.org/citation.cfm?id=2342821.2342823

[3] A. Dimakis, P. Godfrey, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 2000 –2008.

[4] A. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 476 –489, march 2011.

[5] S. Pawar, N. Noorshams, S. El Rouayheb, and K. Ramchandran, "Dress codes for the storage cloud: Simple randomized constructions," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, 31 2011-aug. 5 2011, pp. 2338 –2342.

[6] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with local regeneration," *CoRR*, vol. abs/1211.1932, 2012.

[7] G. M. Kamath, N. Prakash, V. Lalitha, P. Vijay Kumar, N. Silberstein, A. S. Rawat, O. Ozan Koyluoglu, and S. Vishwanath, "Explicit MBR All-Symbol Locality Codes," *ArXiv e-prints*, Feb. 2013.

[8] M. K. Gupta, A. Agrawal, and D. Yadav, "On weak dress codes for cloud storage," *CoRR*, vol. abs/arXiv/1302.3681, 2013.

[9] S. El Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, Oct. 2010, pp. 1510 –1517.

[10] J. C. Koo and J. T. G. III, "Scalable constructions of fractional repetition codes in distributed storage systems," *CoRR*, vol. abs/1102.3493, 2011.

[11] O. Olmez and A. Ramamoorthy, "Repairable replication-based storage systems using resolvable designs," *CoRR*, vol. abs/1210.2110, 2012.

[12] Y. Wang and X. Wang, "A fast repair code based on regular graphs for distributed storage systems," in *Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International*, may 2012, pp. 2486 –2489.